Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift

批量归一化：通过减少内部协变量转移加速深度网络训练

翻译：Kenny(初)，管枫(复)，任远航(审)

绪论：

在深度神经网络的训练过程中，先前层参数的调整会导致之后每一层输入值的分布发生变化，这种现象使模型的训练变得很复杂。所以在深度神经网络模型的训练中，通常需要仔细选取初始参数并采取较小的学习率，这不但导致模型训练的效率低下，而且使得饱和非线性模型的训练极为困难。我们把这种现象称为内部协变量转移(covariate shift)，并通过归一化(normalizing)每层的输入来解决这个问题。我们方法的强大之处在于把归一化的步骤作为模型训练架构的一部分来实现, 并且对每个训练小批量都执行归一化操作。批量归一化允许我们使用很高的学习率并且对初始化不太在意。它在一定情况下也可以起到正则化的作用，并减轻了对Dropout的需求。我们在最先进的图像分类模型中使用批量归一化法，在减少了14倍训练步骤的情况下实现了与原模型相同的精度，并以显著增量击败了原始模型。我们使用批量归一化的网络模型，增强了在ImageNet分类上发布的最佳结果:获得了4.9%前5验证误差（和4.8%测试误差），这超出了人类评估者的准确率。

简介：

深度学习极大地提升了视觉，语言和许多其他领域。随机梯度下降法（SGD）已经被证明是训练神经网络的一个有效的方法，并且随机梯度下降法的变种方法如动量和Adagrad都曾被用来~~去~~获得深度学习的最优性能。随机梯度下降法优化网络的参数，以便最小化损失



其中是训练集。随机梯度下降法是逐步对损失函数进行优化的，在每一步中我们只考虑一个大小为的小批量，并使用



来~~去~~估计损失函数对于参数的梯度。使用小批量样本进行计算，而不是一次计算一个样本，有若干优点。首先，小批量上的损失梯度可以作为训练集上梯度的一个估计，其质量随着批量大小的增加而提高。第二，由于现代计算平台提供的并行性，对一个批量的单次计算比对单个样本的次计算更有效。

虽然随机梯度简单有效，但是它对模型超参数选取的要求较高，特别是优化中的学习率以及模型参数的初始值。实际上，由于每层的输入都会受所有先前层的参数影响，所以网络层数越深，就越容易受到初始参数微小变化的影响。

到整个学习系统的输入分布之改变会影响训练的表现，这被称为协变移位，通常通过 domain adaptation 来处理。然而协变移位的概念可以从针对整个学习系统推广到只针对学习系统的一部分，比如系统的子网络或者一个层。实际上一个层的输入分布发生变化，会使得这个层需要不断的适应新的分布，这也是训练中的一个问题。如果一个网络的损失函数为



其中和可以是任意变换，网络要通过学习参数来最小化损失函数。此时对的学习可以被视做对输入为，并以



为损失函数的子网络的学习。

例如，一个梯度下降步骤



（批量大小为m学习率）完全等同于一个输入为的独立网络。因此，可以使得学习系统训练更有效的那些输入分布的属性—比如在训练和测试数据之间有相同的分布—也适用于子网络的训练。因此的分布随时间保持不变是很有利的，这种情况下不必持续重新调整以补偿分布的变化。

固定一个子网络输入的分布对子网络外的层也会产生积极的影响。考虑一个使用sigmoid激活函数的层，其中是层输入，权重矩阵和向量是学习的层参数，而 。随着增加，趋向于0.这意味着对于的所有维度，除了那些具有小的绝对值的，传导到的梯度会消失并且模型训练会变慢。但是因为是受，和下面所有层参数的影响，在训练期间对这些参数的改变将可能将的许多维度移动到非线性的饱和状态并且收敛减慢。这种效果是随着网络的深度的增加而放大的。在实际应用中，饱和问题（saturation problem）和导致的消失梯度通常通过使用Rectified Linear Units（ReLU）来解决。，仔细的初始化和小的学习率。然而，如果我们可以确保非线性输入的分布在网络训练时保更加稳定，那么优化在饱和状态中停滞的可能会减少，从而加速模型的训练。

我们将训练过程中深度网络内部节点分布的变化称为内部协变量转移，消除它可以使训练加快，对此我们提出了一个新的机制—批量归一化，它将减少内部协变量转移，这样做可以大大地加快深度神经网络的训练。这一机制的实现是通过一个能够固定层输入的平均值和方差的归一化步骤。通过减少梯度对参数规模或其初始值的依赖性，批量归一化还对网络的梯度流动具有有效的效果，这使得我们可以在不提高模型发散的风险的情况下使用更高的学习率。此外，批量归一化正则化模型可以减少对Dropout的需求。最后，通过防止网络陷入饱和模式使得批量归一化可以使用饱和非线性。

在4.2节，我们将批量归一化运用到性能最佳的ImageNet分类网络，结果表明我们可以只使用7％的训练步骤去匹配其性能，并且可以进一步大幅度的超过其精确度。使用~~用~~批量归一化训练的这种网络集合，我们可以获得前5的误差率，它增强了在ImageNet上已知的最佳结果。

2 减少内部协变量转移

我们把在训练期间由于网络参数的变化而造成的网络激活函数输出值分布的变化称为定义内部协变量转移。为了增强训练，我们要寻求减少内部协变量转移。我们期待通过在训练过程中保持层输入的分布来提高训练速度。众所周知如果层输入被白化(whitened)，也就是说把层输入线性变换为零均值和单位方差并且去相关，则网络训练就会收敛得更快。由于每层的输入是由下面层产生的输出，因此对每层输入进行相同程度的白化将是有利的。通过白化每层输入，我们就可以向实现输入的固定分布，并消除内部协变量转移的不良影响的目标前进一步。

我们可以考虑对每个训练步骤或者以一定间隔的激活函数进行白化，也可以通过直接修改网络或者根据网络激活值改变优化算法的参数。但是，如果仅仅将这些修改与优化步骤直接穿插摆放，则梯度下降的步骤对参数的调整可能会改变激活输出的分布并导致重新归一化，而这有可能会使得梯度下降的效果减弱。比如，考虑一个层，输入是加上学习偏置，并且通过减去在训练数据上计算的激活的平均值来对结果进行归一化：其中，是训练集上值的集合，。如果一个梯度下降步骤忽略了对的依赖性，则它更新的值就是，其中。然后。因此，对b的更新和随后的归一化中的变化这两者的组合导致层的输出没有改变，所以也不会改变损失函数。随着训练继续，将无限增长，而损失函数则保持固定不变。如果归一化不仅中心而且缩放激活，这个问题可能变得更糟。我们在初始试验中观察到，当归一化参数在梯度下降步骤外计算时模型就会因为参数发散而不收敛。

上述方法的问题是梯度下降优化没有考虑归一化发生的事实。为了解决这个问题，我们要确保对于任何参数值网络都会产生具有期望分布的激活。这样做能允许相应模型参数的损失梯度考虑归一化以及其对模型参数的依赖性。再次让为层输入，把它视为一个向量，是训练数据集上这些输入的集合。可以将归一化写成一个转换



这不仅取决于给定的训练样本，而取决于所有样本，如果是由另一层产生的，则中的每一个都取决于。对于反向传播算法（backpropagation），我们需要计算雅可比（Jacobians）

；

忽略后一项导致上述参数发散。在这个框架内，白化层输入代价非常大，它需要通过计算协方差矩阵和它的平方根倒数，来给出白化的激活函数输出，并且需要计算上述变换的导数来满足反向传播算法的要求。这促使我们去寻找一种归一化的替代方案，它需要光滑可微，并且不需要在每个参数更新之后对整个训练集进行计算。

一些以前的方法使用在单个训练样本上计算的统计量，或者在图像网络情况下，在一个给定位置上的不同特征。但是，丢弃激活的绝对标量会改变网络的表示能力。相对于整个训练数据的统计，我们想要通过在一个训练样本里归一化激活来保存网络中的信息。

3 通过小批量统计归一化

由于每层的输入完全白化代价太大，并且不是处处可微，所以我们做两个必要简化。第一个是代替白化层的输入的特征和输出，我们将归一化每一个标量特征使得其均值为0方差为1.对于一个d维层输入，我们将归一化每个维度



其中期望值和方差实在训练数据集上计算的。如(LeCun et al., 1998b)所示，归一化加速收敛，即使当这些特征不是去相关的。

注意简单的归一化层的每一个输入可以改变层表达的内容。比如，归一化sigmoid的输入可以约束它们为非线性的线性方程。为了解决上诉问题，我们要确保插入在网络中的转换可以代表特性转换。为了完成这个，我们介绍对于每个激活，用两个参数，缩放和偏移归一化值：



这些参数与原模型参数一起学习，并恢复网络的表示能力。事实上，如果是理想状态下可以通过设置，恢复原始激活。每个训练步骤中的批量设置都是基于整个训练集，我们可以使用整个集合去归一化激活。但是使用随机优化是不切实际的。因此，我们做第二个简化：由于我们在随机梯度训练中使用小批量，每个小批量产生每个激活的均值和方差的估计值。在这种情况下，用于归一化的统计可以完全参与梯度反向传播算法。注意通过计算每个维度的方差而不是联合协方差可以使用小批量；在联合情况中，由于小批量的大小可能小于被白化的激活的数量，导致奇异协方差矩阵的产生，所以可能需要正则化。

考虑到一个小批量的大小是。由于归一化被独立的运用到每个激活，为了清楚起见，我们把注意力放在一个特定激活并且忽略。经济在小批量中对于这个激活我们有个值，



令归一化值为，他们对应的线性转换是。我们把变换



作为批量归一化（BN）转换。我们在算法1中提出BN转换。在这个算法里是用于数值稳定性添加到小批量方差的常数。

|  |
| --- |
| **输入:** 在一个小批量上的值：；  学习参数是 |
| **输出**： |
| //批量均值 |
| //批量方差 |
| //归一化 |
| //缩放和偏移 |

算法1：BN转换，应用于小批量上的激活。

BN转换可以被添加到网络上去操作任何激活。在里，我们表明是学习参数，但是应该注意的是批量转换不独立处理每个训练样本中的激活。相反，依赖于训练样本和小批量中的其他样本。这个缩放和偏移值被传递到其他网络层。这个归一化的激活是我们转换的内部，但它们的存在是至关重要的。只要每个小批量的元素是从相同分布采样，如果我们忽略，任何的值的分布都具有期望值为0和方差为1。这个可以通过观察和和采取期望看到。每个归一化的激活可以被视为由线性转换组成的一个子网络的输入，其次是由原始网络进行其他处理。这些子网络的输入全部都被固定均值和方差，而且尽管这些归一化的的联合分布可以在在训练过程中改变，但是我们期望引入归一化的输入来加速子网络的训练，因此将网络作为一个整体。

在训练过程中我们需要反向传播算法损失的梯度，通过过这种转换，以及计算BN转换的相关的参数的梯度。我们是用链规则如下所示（在简化之前）：













因此BN转换是将归一化激活引入网络的可微分转换。这确保了当模型在训练时，层可以继续对表现出较少内部协变量转移的输入分布进行学习，从而加速训练。此外，将学习的仿射转换应用到这些归一化的激活中可以允许BN转换代表身份转换并保留网络容量。

3.1训练和推理批量归一化网络

批量归一化一个网络，我们制定一个激活的子集，根据算法1为子集中的每一个插入BN转换。任何层由先前的接收作为输入到现在接收作为输入。可以使用批量梯度下降，一个尺寸的小批量随机梯度下降，或者它的任何一个变体比如Adagrad来训练一个批量归一化模型。依赖小批量的激活的归一化可以有效地训练，但是在推理期间就没有必要也不要期望这种效率。我们想要输入决定输出，对此，一旦网络被训练时我们就要使用归一化



使用population,而不是小批量，统计。忽略，在训练时这些归一化的激活有相同的均值0和方差1。我们使用无偏差方差估计,其中期望和方差是针对训练大小为的小批量。当它训练时我们可以追踪模型的精确性来代替使用移动平均线。由于在推理时，均值和方差都是固定不变的，归一化仅仅是应用于每个激活的线性转换。这个还可以用缩放和用偏移代替产生一个单线性转换。算法2总结了训练批量归一化网络的过程。

|  |
| --- |
| **输入**：具有可训练参数的网络  激活的子集 |
| **输出：**用于推理的批量归一化网络， |
| 1： //训练BN网络 |
| 2：for  do |
| 3: 增加转换到 |
| 4: 使用输入代替修改中的每一层 |
| 5: end for |
| 6: 训练去优化参数 |
| 7:  //用冻结的参数推理BN网络 |
| 8: for  do |
| 9： //为了简单明了， |
| 10： 处理多个训练小批量,每个小批量大小为,对他们求均值和方差 |
| 11： 在里，用代替 |
| 12：end for |

算法2:训练一个批量归一化网络

3.2 批量归一化卷积网络

批量归一化以应用于网络中激活的任何集合。这里，我们关注的转换包括一个仿射和一元非线性：



其中和是模型的学习参数，是非线性函数，比如Sigmoid或者ReLU。这个公式涵盖全连接层和卷积层。我们通过归一化在非线性之前加上BN转换。我们也可以归一化层输入，但是由于可能是另一个非线性的输出，其分布的形状可能在训练期间改变，并且约束其第一和第二矩将不会消除协变量转移。相反的，更更可能具有对称，非稀疏分布，即“更高斯”；归一化它可能产生具有稳定分布的激活。

注意，偏置b可以被忽略，因为他会被随后均值减去（根据算法1的带入偏置的作用），由于我们归一化，因此，是可以被取代的。其中BN转换被独立的应用到的每个维度，每个维度都具有单独的一对学习参数。

对于卷积层，我们还希望归一化服从卷积属性—使得相同特征图的不同元素在不同位置处以相同方式归一化。为了实现这个，我们联合归一化了小批量中所有所有位置的所有激活。在算法1中，我们让B是一个特征映射中的所有值的集合，包含了小批量和空间位置的元素—因此对于大小为的小批量和大小为的特征映射，我们使用大小为的有效小批量。我们对每个特征映射学习一对参数而不是每个激活。算法2也是被类似修改，使得在推理期间BN转换将相同的线性变换应用在一个给定特征映射中的每个激活。

3.3批量归一化能够使用高的学习率

在传统深度网络中，过高的学习率可能会导致梯度过拟合或者消失，以及陷入不好的局部最小值。批量归一化可以解决这个问题。通过归一化整个网络中的激活，其防止参数的小变化扩大到大变化和梯度中激活的次优变化；比如它阻止了训练陷入非线性的饱和状态。

批量归一化还使训练更加适应参数规模。通常，大的学习率可能增加层参数的规模，然后在反向传播算法期间放大梯度并导致模型爆炸。但是，在批量归一化下，一个层的反向传播算法是不受它的参数规模影响。事实上，对于一个标量，







这个规模不影响层雅可比，也不影响梯度传播。此外，较大的权重导致较小的梯度，并且批量归一化将使参数稳定增长。我们进一步推测，批量归一化可以使层雅可比矩阵具有接近1的奇异值，众所周知这是有益于训练的。考虑到具有归一化输入的两个连续层以及这些归一化向量之间的转换。如果我们假设和是高斯并且不相关的，给定模型参数的一个线性转换，然后和都有单位协方差，并且。因此,所以的所有奇异值都等于1，这样可以在反向传播算法期间保存梯度大小。实际上，转换不是线性的，并且不能保证归一化的值是高斯的也不是独立的，但是我们希望批量归一化帮助梯度传播执行的更好。批量归一化对梯度传播的精确作用仍然有进一步研究的领域。

3.4 批量归一化可以正则化模型

当使用批处理标准化进行训练时，结合小批量中的其他样本来看训练样本，训练网络不再为一个给定的训练样本产生确定的值。在我们的实验中，我们发现这种效果有利于网络的泛化。而Dropout通常用于减少过拟合，在批量归一化网络中，我们发现它可以被去除或降低过拟合的强度。

4 实验

4.1 随时间的激活

验证内部协变量转移对训练的影响，以及批量归一化对抗它的能力，我们考虑了在MNIST数据集上预测数字类别的问题。我们使用一个非常简单的网络，用一个28\*28二进制图像作为输入，3个全连接隐藏层，每层100个激活神经元，每个隐藏层用sigmoid非线性计算，权重W被初始化为小的随机高斯值。最后一个隐藏层后面是一个有10个激活的完全连接层（一个表表一个类）和交叉熵损失，我们设置训练网络为50000步，每个小批量有60个样本。我们向网络的每个隐藏层添加了批量归一化，如3.1节所示。我们对基线和批量归一化的网络之间的对比感兴趣，而不是在MNIST上实现的最先进的性能。



图1：（a）MNIST分别在有BN和没有BN训练的网络上的测试准确性，对比在不同数量的训练步骤。批量归一化网络帮助网络训练更快并且获得更高的准确性。（b,c）在训练过程中，一个典型的sigmoid输入分布的演变，显示为{15%，50%，85%}。批量归一化使得分布更加稳定并且减少了内部协变量转移。

图1（a）显示的是随着训练的进展，两个网络对互斥测试数据的正确预测的分数。批量归一化的网络测试准确性比较高。为了研究为什么，在训练过程中，我们在原始网络N和批量归一化网络中研究了S形的输入（算法2）。在图1中，我们显示，对于来自每个网络的最后一个隐藏层的一个典型激活，其分布如何演变。原始网络中的分布随时间在它们的平均值和方差中改变显著，这使的后续层的训练变得复杂。相比之下，批处理归一化网络中的分布随着训练进展而更加稳定，这是有助于训练的。

4.2 ImageNet **分类**

我们将批量标准化应用于Inception网络的一个新变体，在ImageNet分类任务上进行了训练。这个网络有大量的卷积和池化层，一个预测图像类别超出1000个可能性的softmax层。卷积层使用ReLU作为非线性。与(Szegedy et al., 2014)描述的网络主要的区别是5×5的卷积层由具有多达128个滤波器的3×3的两个连续卷积层代替。这个网络包含13.6\*106个参数，除顶层softmax层之外，没有全连接层。更多的细节在附录中给出。我们在文章的其余部分将这个模型称为Inception。使用大小为32的小批量和具有动量的随机梯度下降的版本来训练模型。使用大规模的分布式架构进行训练(类似于 (Dean et al., 2012)).在训练过程中通过计算验证准确性@1来评估所有网络，即在互斥集上使用每个图像的单个裁剪来预测1000个可能性中的正确标签的概率。

在我们的实验中，我们评估了归一化的Inception的几个修改。在所有情况下，批量归一化应用于每个非线性的输入，以卷积方式（如3.2节所描述），同时保持架构的其余部分不变。

4.2.1 加速BN网络

简单地将Batch Normalization添加到网络并不能充分利用我们的方法。为此，我们进一步改变了网络及其训练参数，如下：

增大学习率。在一个批量归一化模型，我们已经能够在较高的学习率下实现训练加速，而且没有不良的副作用（3.3节）。

去除Dropout。如3.4节所描述，批量归一化满足与Dropout相同的目标。从修改的BN-Inception中去除Dropout可加快训练，而切不会增加过拟合。

减少权重正则化。在Inception里，模型参数上的损失控制过拟合，在修正的BN-Inception中，这个损失的权重减少了5倍。我们发现这提高了互斥验证数据的准确性。

加速学习速率衰减。在训练Inception，学习率以指数方式衰减。因为我们的网络训练比Inception快，所以我们将学习速度降低了6倍。

去除本地响应归一化当Inception和其他网络(Srivastava et al., 2014)从中受益时，我们发现使用批量归一化没有必要。

更彻底地改组训练样本。我们启用了训练数据的内部改组，这防止了相同的样本总是出现在一个小批量中。这导致验证准确性提高约1％，这与批量归一化作为一个正则化的观点一致（3.4节）：当每次看到它不同的影响一个样本，我们的方法中固有的随机化应该是最有益的。

减少光度的扭曲。因为批量归一化网络训练更快，观察每个训练样本的次数更少，所以我们通过更少的扭曲关注更多的“真实”的图像。